END

FILMED

DTIC

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AD-A152 269

REPORT NO. ADCR-85-1
Volume II

ESP — A PILOT COMPUTER PROGRAM FOR
DETERMINING FLUTTER-CRITICAL
EXTERNAL-STORE CONFIGURATIONS

VOLUME II - FINAL REPORT ON PROGRAM ENHANCEMENT AND DELIVERY

February 1985

Prepared Under Contracts N00019-81-C-0395
and N00019-84-C-0123

JOHN B. SMEDFJELD

GRUMMAN AEROSPACE CORPORATION
BETHPAGE, NEW YORK 11714

DTIC
S ELECTE D
APR 0 8 1985
E

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

85    4   04   04

# ESP - A PILOT COMPUTER FOR DETERMINING
## FLUTTER-CRITICAL EXTERNAL-STORE CONFIGURATIONS

Volume II - Final Report on Program Enhancement and Delivery

John B. Smedfjeld

February 1985

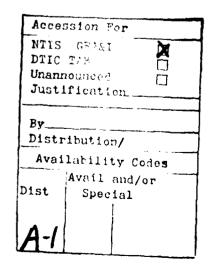FOREWORD

The report consists of three volumes. Volume I, entitled "User's Manual", provides instructions for using the ESP program and presents descriptions of typical output. Volume II, "Final Report on Program Enhancement and Delivery", describes the work that was performed under the two contracts. A listing from a CDC compilation of the program is contained in Volume III, "Program Compilation".

# CONTENTS

Accession For

NTIS  GRA&I        X
DTIC TAB           ☐
Unannounced        ☐
Justification_____

By_____
Distribution/

Availability Codes

Dist | Avail and/or
     | Special

A-1

iii

# 1 - SUMMARY

A pilot computer program for determining flutter-critical external-store configurations, which was partially developed under two previous Naval Air Systems Command contracts, has been further enhanced and made operational on the Central Computer System at the Naval Air Development Center. The enhancements were introduced to:

(1)  Increase the applicability of the program to modern attack aircraft with thinner, more flexible wings, and

(2)  Permit the utilization of aircraft dynamics-model data from both the COSMIC and the MacNeal-Schwendler Corporation versions of NASTRAN, and also from card-image files.

Additional modifications, including a substantial reduction in computer core requirements, were made to convert the original IBM code to a CDC version compatible with the computing facility at NADC. Also, logic was added to by-pass data required for store-search runs if only a conventional flutter analysis is desired. Finally, a user's manual was prepared (Volume I of this report), which, in conjunction with a FASTOP user's manual and previous theoretical documentation of the store-search procedure, should permit a practicing flutter analyst to use the new External-Stores Program (ESP) effectively.

## 2 - INTRODUCTION

During the development of the initial version of the Flutter And STrength Optimization Program (FASTOP), Reference 1, under contract F33615-72-C-1101 from the Air Force Flight Dynamics Laboratory, Mr. Keith Wilkinson, the project engineer on that contract, recognized that much of the technology being used for minimum-weight structural resizing in FASTOP also had the potential for substantially reducing the time and cost required to determine which combinations of wing-mounted external stores would result in the lowest aircraft flutter speeds.

Subsequently, under contract N00019-76-C-0160 from the Naval Air Systems Command, as well as a complementary Grumman Independent Research and Development project, a search algorithm for wing/store flutter was developed, refined, and tested by modifying and expanding the FASTOP code (see References 2 and 3). When this development effort led to a pilot program that exhibited both good reliability (absence of search failure) and good convergence characteristics, work was continued under a second NASC contract, N00019-79-C-0062, to add features desirable for practical applications and to demonstrate the new External-Stores Program (ESP) on a representative attack aircraft and its associated store inventory (see Reference 4). The project engineer on these studies was Mr. Richard Chipman.

With the performance and the advantages of the store search procedure having been confirmed, utilization of the procedure on current aircraft projects became desirable. This volume describes work that was done to permit early availability of ESP to practicing flutter analysts, to increase the applicability of the program to modern attack aircraft, and to permit dynamics-model data to be read into ESP directly from NASTRAN output files or from card-image files.

# 3 - DISCUSSION

## 3.1 - BACKGROUND

At the conclusion of the study described in Reference 4, a pilot External-Stores Program (ESP) had been developed on Grumman's IBM computing facility, and the capabilities and advantages of the new program had been demonstrated by applying it to the A-6E aircraft and its extensive store inventory.

However, for newer attack aircraft designs, improvements in the vibration- and flutter-analysis capabilities of ESP were judged to be desirable to permit realistic and efficient analyses and store-search calculations for these aircraft.

One important design trend that created a need for improved capabilities is the use of thinner wings, and hence more flexible wing structures. The resulting lower vibration-mode frequencies can lead to additional and more-complex flutter mechanisms, especially with wing-mounted external stores. This, in turn, creates a requirement for flutter analyses based on a larger number of modes.

The reduction in vibration-mode frequencies due to thinner wings also has the effect of increasing the coupling between the structural modes and the rigid-body modes of an aircraft, again especially for wings with external stores. Therefore, for modern attack aircraft, capability should exist for readily introducing rigid-body modes in a flutter analysis.

Associated with the inclusion of rigid-body modes are two additional requirements for ESP enhancements. First, the range of reduced frequencies for which generalized aerodynamic forces are to be determined must be greatly expanded. Also, provision must be made for the limiting case of zero-frequency modes.

Another design feature of newer aircraft that creates a need for an increased number of modes is the more sophisticated use of control surfaces.

This increase is in part due to the larger number of control surfaces being used. However, a more significant influence is the trend toward using control devices at wing leading edges. This practice introduces vibration modes that, although having high natural frequencies at zero airspeed, exhibit substantial declines in frequency with increasing airspeed, and thus can produce significant coupling effects with lower-frequency modes.

Although increasing the number of modes used in a flutter analysis may be important for realistic mathematical modelling of modern attack aircraft, such increases must be made judiciously, since computer-time requirements can become large. The p-k flutter-solution method, which is an essential ingredient in the store-search procedure in ESP, requires a significant percentage of the total computer-time usage for a typical search step, and the time required for this solution is proportional to the cube of the number of modes. Therefore, a means for automatically eliminating, prior to each p-k solution, any modes that should not significantly affect the flutter characteristics would be highly desirable.

The discussion thus far has focused on some enhancements to ESP that were deemed to be important based on technical or computer-time considerations. However, certain additional modifications were seen to be advantageous for increased flexibility of usage and to make the program more generally available to the flutter analysis community.

When the pilot ESP code discussed in Reference 4 was developed from the FASTOP code documented in Reference 1, the use of the finite-element structural-analysis module in FASTOP to provide much of the data needed for the vibration-analysis module was retained. In many situations, however, it would be desirable to utilize ESP after structural-analysis results and possibly other data files had been created via other programs such as NASTRAN. Thus, additional options for specifying vibration-analysis input data should be available.

Also, although the development of ESP was aimed initially only at the demonstration of the store-search procedure, in routine usage it might at times

3-2

be desirable to perform individual flutter analyses using ESP. For such situations, an option should be provided for by-passing input data that is technically necessary only for search runs.

Finally, to make the program readily available at the discretion of the Naval Air Systems Command to persons outside Grumman, it should be made operational on a government computing facility. In addition, an ESP user's manual should be published.

## 3.2 - OBJECTIVES

To satisfy the ESP enhancement requirements that occurred due to the design characteristics of newer attack aircraft, several specific modifications to the ESP pilot code were defined. The maximum number of structural modes that can be calculated in the vibration-analysis module was to be increased from 20 to 40. An option was to be added for appending the rigid-body modes of an aircraft to the structural modes by extracting this information from the transformation matrix from relative to absolute coordinates which was already being used in the vibration-analysis module. The maximum number of total modes (structural plus rigid-body) that could be calculated/assembled in the vibration-analysis module and used in the flutter-analysis module also was to be increased from 20 to 40. Thus, if the inclusion of 3 rigid-body modes is specified by a user, the maximum number of structural modes that could be calculated and used would be 37.

Additionally, an increase from 6 to 15 was to be made in the maximum number of reduced frequencies for which generalized aerodynamic forces are calculated and later used as a base for interpolation. To permit near-optimal coverage in each application, both the number of reduced frequencies and the actual values to be used was to be specified by the user. To accommodate rigid-body modes with no aerodynamic spring, such as heave, the flutter-analysis module was to be modified so that calculations could be performed for frequencies and reduced frequencies very close to zero.

The automatic-mode-elimination feature would be based on ratios of generalized forces to generalized masses. Prior to each p-k flutter solution, generalized aerodynamic forces would be computed for a few velocities and for all the vibration modes which the user has tentatively chosen to consider in the flutter analysis. Then, the modulus of each on-diagonal generalized-force term would be divided by the value of the corresponding generalized-mass term, and any mode for which this ratio is below a user-prescribed cut-off value for all the velocities considered would not be included in the p-k flutter solution.

To satisfy the ESP requirements associated with increased availability and flexibility of usage for persons outside Grumman, additional tasks were defined. Three new types of interfaces were to be established for transferring data to the ESP vibration-analysis module from other programs: a direct transfer from both the COSMIC and the MacNeal-Schwendler versions of NASTRAN, and a transfer via card images that could be generated by a user-provided data-conversion program executed as an intermediate step between another upstream program and ESP. Four dynamics-model matrices would be transferred using the three new interface types: the flexibility matrix, the mass matrix associated with the dynamic degrees of freedom other than those that are assumed to be fixed when computing the flexibility matrix; a separate mass matrix for the degrees of freedom at the assumed free-body support point; and the matrix of displacements in the dynamic degrees of freedom due to unit rigid-body displacements.

As another flexibility-of-usage enhancement, code was to be added for checking the value of an "analysis-only" input clue to determine whether or not data required only for search runs should be read as part of a particular input data file. Also, for increased user availability, the program was to be converted from its original IBM version to run on CDC equipment, and then it was to be installed on the Central Computer System at the Naval Air Development Center. A maximum central-memory usage of $300K_8$ words was established as a goal. Also, maximum commonality was to be maintained between the IBM and the CDC versions to facilitate program maintenance, future enhancements, and possible future use of the IBM version outside Grumman.

The user's manual to be published would take advantage of the considerable commonality between ESP and FASTOP; i.e., to avoid unnecessary bulk, it would refer to, rather than duplicate, the large blocks of FASTOP material that are unchanged in ESP.

3.3 - APPROACH

The work required to implement the program-modification objectives cited above falls naturally into two categories, which will be referred to in the following text as "technical enhancements" and "CDC conversion". Although, for engineering clarity, the vast majority of the text thus far has been devoted to the technical-enhancement tasks, the resources required to accomplish the two types of tasks were expected to be roughly comparable. Considering that the vast majority of the work had to be done in a batch computing environment, it was decided to perform the work in the two categories largely in parallel. It was felt that this would minimize total elapsed time, and also help avoid a possible need to recode some of the technical enhancements in the CDC conversion phase.

More specifically, it was decided to start work at the beginning of Contract N00019-81-C-0395 on the conversion to CDC (including installation at NADC) of the version of the program that existed at the conclusion of the work reported in Reference 4. Then, when this effort was well underway, work was initiated on introducing the technical enhancements into the original IBM version. This approach led to a situation in which the original program was operational at NADC, and a program with many of the technical enhancements was operational on Grumman's IBM facility. The enhancement and conversion updates to this point were then combined into a new IBM and a new CDC version with maximum commonality. Finally, the technical enhancements were completed, and a last minor CDC conversion was performed.

A two-stage approach was used to perform the CDC conversions. Initially, the code was converted from Grumman's IBM facility to Grumman's CDC Cyber 740; then, a tape of the code was sent to NADC for debugging/checkout on the Central Computer System there. This approach had two major advantages: First, the

3-5

vast majority of the CDC debugging could be done on a facility with which many persons at Grumman are familiar. Also, the direct links between all of Grumman's large computer systems, both batch and interactive, as well as an ability to run large jobs during prime time, provided good turnaround.

The approach to obtaining maximum commonality between the IBM and CDC versions of the ESP code extended beyond the practice of using code acceptable to both systems wherever possible. Where differences between the two versions were unavoidable, both versions of the lines of code that were different were included in both versions of the total program, with the inapplicable lines of code for each version being included in the form of comment cards. Additional comment cards were used to designate the beginning and the end of each group of IBM-only and CDC-only code. These cards constituted clues for a Grumman utility program that automatically converts appropriately configured IBM source code to equivalent source code suitable for input to the CDC UPDATE procedure. To illustrate the approach used and the results obtained, a portion of a flutter-analysis plotting routine, FLUTAP, is shown in its IBM and CDC versions in Figures 3-1 and 3-2, respectively.

Modifications to the IBM version of the ESP code, to introduce both the technical enhancements and the new code required for CDC conversion, were made using another Grumman utility program that is similar in concept to the CDC UPDATE procedure in that it provides a record of the changes that are made. The use of this utility not only facilitated methodical program updating, but also contributed future benefits in the areas of troubleshooting, program maintenance, and additional updating.

The development of the dynamics-model interface capability for both COSMIC and MacNeal-Schwendler Corporation (MSC) NASTRAN files, and also for card-image files, was performed under Contract N00019-84-C-0123. A three-stage approach, using NASTRAN data provided by NAVAIR for a realistic sample problem, was followed for the two NASTRAN interfaces. First, the NAVAIR data was modified to write the desired matrices to data files to be passed to ESP. The OUTPUT2 utility routine (see Reference 5, pages 5.5-24 through 5.5-27) was selected for writing the COSMIC files and the OUTPUT4 procedure (see Reference 6, pages

3-6

nfluence-coefficient and p-k flutter-solution portions of the code. However, only minor modifications were required to extend the maximum reduced velocity to a value that was considered acceptable in items of the bias that would be added to a rigid-body modal frequency. Specifically, for a typical attack-aircraft semichord length and a high subsonic speed, it was found that reliable results could be obtained for frequencies as low as 0.005 Hz. These results were obtained with a set of reference reduced velocities in which the highest value was $1 \times 10^5$ and the next highest value was 2000; these upper-range reduced-velocity values are recommended whenever a rigid-body modal frequency less than about 0.01 Hz is specified. To avoid program terminations due to inadvertent user specification of an excessively low rigid-body frequency, code was added to reset input frequencies less than 0.002 Hz to a 0.002 Hz value. No new input-data items or significant increase in central-memory usage was associated with the coding changes for zero-frequency modes.

For the implementation of the "analysis-only" option, use was made of an existing input clue from the FASTOP program that was used to specify whether or not entry to the optimization module was desired. In the new ESP program, this clue also was used to determine whether or not data that is required only for store-search runs is to be read as part of an input file.

3.4.3 - Second CDC Conversion

As work was nearing completion on the various ESP enhancements discussed above, a parallel effort was initiated on making the new code operational on Grumman's CDC facility and, subsequently, at NADC. As a result of the extensive earlier effort on CDC conversion, and also the continuous attention to maintaining a near-constant central-memory usage as the new capabilities were being introduced, the second CDC conversion involved only one substantial additional task: To minimize computer-time expenditures due to the new out-of-core operations, most of the new intermediate disk input/output that had been used in the IBM version was changed to use Extended Core Storage in the CDC version. This modification encompassed the new input/output that was introduced as part of the initial CDC conversion effort, as well as the later changes to the flutter-solution code to accommodate the increased number of

3-20

procedure described in Reference 7 is not well suited to saving and
equently utilizing disk files consisting of multiple arrays. Since the CDC
procedure is not limited in this way, and since the saving of up to 15
vidual AIC files in any given run would be both cumbersome for a user and
ficient in terms of central memory required for buffers, a change was
emented to write all AIC arrays as a single disk file via one input/output
. Since the multiple-array limitation of the Grumman IBM DSIO procedure
not exist for tape storage, commonality between the IBM and CDC versions
he program could be maintained despite the change by simply redefining the
input/output unit as a tape when executing on an IBM facility.

Since one objective associated with the increased number of reference
iced velocities was to permit the actual number of these values utilized in
run to be specified by the user, a new input-data item for this quantity
introduced. No significant increase in central-memory usage and/or inter-
late input/output operations were associated with the increased number of
rence reduced velocities.

Coding changes to accommodate the limiting case of zero-frequency modes
associated primarily with new logic to avoid dealing with a
iced-velocity value of infinity in the generalized-force interpolation
cedure. The approach taken was to implement a two-zone procedure in which
interpolation is with respect to reduced velocity for reduced velocities
than 10, and is with respect to reduced frequency for reduced velocities
iter than 10. Also, the use of the reduced velocity as a weighting factor
he generalized forces prior to interpolation (as shown on page 84, Volume
Reference 1) was changed so that no weighting occurs when the interpolation
ment is a reduced velocity greater than 10. With this two-zone approach,
alues of the independent or dependent variables in the interpolation
cdure approach infinity for either very high or very low reduced
cities.

Trial calculations with the new interpolation procedure showed that the
mum reduced velocity for which the program could be used was still con-
ined to a significant extent by additional limitations in the aerodynamic-

3-19

already existing code. Non-zero rigid-body generalized-spring values, if any, were obtained by combining the user-supplied rigid-body frequencies and the previously computed generalized masses. Since, in general, the reference point for the relative-coordinate system will not be at the airplane center of gravity, provision was included for computing off-diagonal as well as diagonal rigid-body generalized-spring terms (see Appendix B in Volume I of this report). It is this calculation that makes use of the new input-data items cited above that specify the total number of rigid-body modes to be used in the flutter analysis, and the number of these that are translation modes. No significant increase in central-memory usage and/or intermediate input/output operations were associated with the new capability for including rigid-body modes in a flutter analysis.

The increase in the maximum number of reference reduced velocities required two types of program modifications in addition to the obvious increases in the dimensions of certain arrays. One of these was a natural extension of the elimination of the generalized-force-interpolation accuracy test discussed above. Inherent in the use of this test was logic specifying that the generalized forces be calculated for the six reference reduced velocities in a nonsequential order, viz., 1, 2, 3, 6, 4, 5. With the accuracy test eliminated, and with the number of modes and reference reduced velocities being increased to 40 and 15, respectively, it was not only unnecessarily awkward to retain the original generalized-force-calculation order, but the required subsequent reordering of the generalized forces, in preparation for the generalized-force interpolations during the flutter-solution procedure, was becoming a significant user of computing time. Therefore, the gener- alized-force-calculation order was changed from that cited above to a direct ascending sequence according to the specified reduced velocities.

The second type of logic change associated with the increase in the number of reference reduced velocities concerned the method of saving aerodynamic influence coefficients (AIC's). In the version of ESP used for the demonstration study reported in Reference 4, the AIC array for each reduced velocity was written as a separate disk file via a different input/output unit. This approach was motivated by the fact that the Grumman IBM equivalent of the

Trial calculations with the procedure for automatic mode elimination showed that its use could influence significantly the path of a search, even when the eliminated modes were of relatively high order and had an insignificant influence on the flutter speed. This behavior can be attributed to the fact that derivative calculations are much more sensitive than the flutter characteristics themselves to the system-idealization changes resulting from the mode eliminations. Therefore, caution is advised when using the automatic-mode-elimination option in conjunction with store-parameter search runs.

The addition of an option for including rigid-body modes in a flutter analysis involved the introduction of the following new input-data items: a clue to specify whether or not this option is to be utilized; the number of rigid-body degrees of freedom that are being used in the vibration analysis; the number of rigid-body modes to be used in the flutter analysis; the number of rigid-body flutter-analysis modes that are translation modes; and the zero-airspeed frequencies that are to be assumed for the rigid-body modes. The ability to specify nonzero rigid-body frequencies at zero airspeed, although physically a self-contradiction, was introduced to permit a user to include to some limited degree effects that are not represented well or at all by the present capabilities of the program. Examples of these effects are aerodynamics of nonconventional fuselage geometries (such as that of the F-14), and modifications to rigid-body dynamic characteristics due to a flight-control system (especially important for control-configured vehicles). With the capability to specify a zero-airspeed rigid-body modal frequency being added to the previous capability to specify a zero-airspeed rigid-body modal damping value, a better total idealization for flutter-analysis purposes may be possible in some situations with little user effort.

As has been mentioned previously, the rigid-body modes themselves were obtained from the transformation matrix from relative to absolute coordinates already being used in the ESP vibration-analysis module. The rigid-body modes were assigned indices from one to the number of these modes (maximum of three, assuming either symmetric or antisymmetric motion), and the indices of the flexible modes were shifted upward accordingly. The rigid-body generalized masses were then computed along with those for the flexible modes using the

of this array was easily accomplished. However, with the larger number of modes, central-memory usage with this approach became excessive, and, accordingly, an out-of-core transpose became necessary. Further, for the generalized-force-interpolation accuracy test, many calls to the interpolation routine with small arrays, as discussed above, became necessary. At this point, it was decided to eliminate the generalized-force-interpolation accuracy test, since, for several years, this had been considered superfluous by FASTOP users at Grumman. (The commonly used approach in this area had been to set the interpolation tolerance to a very small number, so that six reference reduced velocities would always be used.) With this simplification, a portion of the computing-time increase associated with the changes just discussed was avoided, and, in addition, some of the original calculation time was eliminated.

The addition of logic for the automatic elimination of modes from the p-k flutter-solution procedure, based on ratios of on-diagonal generalized forces to corresponding generalized masses, involved the addition of three new input-data items: a clue to specify whether or not this option was to be invoked; a value for the above ratio which would be used as the criterion for the mode elimination; and a nominal velocity at which the generalized forces to be used in the calculated ratios for the various modes would be determined. A new routine was written in which this information was used to calculate the generalized-force/generalized-mass ratios at three velocities (0.75, 1.0, and 1.25 times the nominal value) and then to determine which modes are to be eliminated. The reduced velocities at which the generalized forces were calculated (via interpolation) were determined based on the zero-airspeed modal frequencies. The results from the automatic-mode-elimination calculation were added to the printed output.

The actual elimination of the modes prior to the flutter solution was carried out using a routine previously used for the elimination of modes based on direct user specification via the input data on page 237, Volume II, Reference 1. The elimination of the modes in the flutter-speed-derivative calculation was accomplished using a new routine tailored to the structure of the modal data in that portion of the program. To accommodate the automatic-mode-elimination update without increasing central-memory requirements, a small amount of additional intermediate input/output was introduced.

3-16

specified in Subsection 3.2 into the IBM version. Increasing the maximum
number of modes from 20 to 40 was addressed first, since this was expected to
be the most difficult of the enhancements due to its substantial impact on
central-memory requirements. A straight-forward increase in dimensions within
the various affected subroutines and common blocks resulted in an increase of
approximately 50% in central-memory usage, and a subsequent reassessment of the
overlay structure with the new dimensions led to the conclusion that only a
small saving could be achieved by a further rearrangement. Further, there were
several segments that extended well beyond the range of memory usage that was
established as a maximum.

One of the major technical areas in the program that was responsible for
the increased memory usage with the larger number of modes was the
flutter-solution package. Both the k and p-k methods, including the associated
generalized-force interpolation, had experienced a large growth, since these
methods used many moderate-to-large arrays that had been increased in size by a
factor of either 2 or 4. Two types of modifications were made to again reduce
the central-memory usage in this area to a level equivalent to less than $300K_8$
words on CDC equipment: New intermediate input/output operations were
introduced to permit several arrays to occupy the same memory area at different
times in the solution; also, the previously existing intermediate input/output
associated with the generalized-force interpolation was changed to be performed
via numerous read/write executions involving small arrays rather than one
execution for a large array.

A second major area of the program requiring attention due to the in-
creased number of modes was the concluding portion of the generalized-force
calculation procedure. This area performed the generalized-force-interpolation
accuracy test discussed on pages 88-91, Volume I, and pages 227-228, Volume II,
of Reference 1, and also collected all the elements of the generalized-force
arrays for all the reduced velocities into a form suitable for input to the
generalized-force interpolation associated with the flutter-solution proce-
dures.

Originally, the entire collection of generalized-force elements had been
allocated space in central memory, and therefore a required implicit transpose

3-15

A few other changes were required to convert ESP to CDC equipment because the version of FASTOP that formed the basis for the initial development of ESP was not the final checked-out version delivered to the Air Force in conjunction with Reference 1. Some of the updates that were made later to FASTOP also were made to ESP, since these updates were essential to the IBM-version development and demonstration work reported in References 2 through 4. However, ESP updates uniquely associated with a CDC version were not required nor made until this contract.

Check-out of the modifications to reduce central-memory requirements and to convert to CDC equipment was accomplished primarily by re-executing portions of typical search runs made during the study reported in Reference 4. These included both the generation and the later utilization of aerodynamic influence coefficients from the subsonic doublet-lattice procedure. Being searches, which require root tracking and automatic determination of the minimum flutter speed, only the p-k flutter-solution procedure was exercised in these runs. In complementary runs, the k-method flutter-solution procedure was also checked. CalComp plots of the flutter solutions were obtained from both methods.

Following the check-out on Grumman's CDC facility of the version of ESP without the technical enhancements, a tape of the source code, the segmentation deck, and sample data sets was sent to NADC. Due to significant differences between the Grumman and NADC CDC operating systems, progress in getting ESP operational on the NADC Central Computer System was initially slow. However, with assistance from NADC personnel, successful operation, including CalComp plotting, was achieved following only a few coding changes. It was subsequently determined that these changes were acceptable on the Grumman CDC facility as well. Thus, at this stage, complete commonality between the Grumman and NADC CDC source codes and segmentation decks was possible. Only the control cards were different.

3.4.2 - Technical Enhancements

Shortly after the start of the CDC-conversion effort described thus far, work was also initiated on the introduction of the technical enhancements

3-14

To complete the initial transfer of ESP to the Grumman CDC facility, two additional steps were required. First, two subroutines that had been obtained from a mathematical library in Grumman's IBM system were added in source-code form to ESP. Also, a CDC segmentation deck was developed corresponding to the IBM overlay deck.

The initial emphasis in making ESP operational on the CDC facility was in the area of DSIO usage. As noted in Reference 7, the CDC input/output units associated with Fortran READ/WRITE statements are independent of the DSIO units, i.e., the same unit numbers can be used with both input/output types. Also, the buffer space associated with DSIO is defined by a two-dimensional array having as its dimensions the buffer size for each unit and the highest unit number. Thus, for central-memory conservation, all CDC DSIO unit numbers should ideally form a contiguous array from 1 to the maximum number of units used. On an IBM facility, however, the DSIO unit numbers must be different from any Fortran unit numbers, and it is not important that they form a contiguous set. Therefore, in the ESP code that was not previously in FASTOP, the DSIO usage had to be recoded for the CDC version. In the process, some IBM unit numbers were also changed to achieve maximum commonality. Specifically, as in the previous FASTOP code, the IBM unit numbers were selected to be the same as the CDC numbers except for a gap of three numbers to allow for the traditional Fortran reading, writing, and punching on units 5, 6, and 7, respectively.

Another area requiring coding changes to achieve operational status on the CDC facility was the generation of CalComp plots of the flutter-analysis solutions. Although this area of code was not uniquely associated with ESP, it had existed only in IBM versions of FASTOP. The changes introduced here for the CDC version were primarily associated with CalComp subroutine argument differences between IBM and CDC, buffer-size differences, and the introduction of a CDC input/output unit to receive the output for the plotter. (This plotter output unit was not required in either Grumman's IBM system or Grumman's CDC system, because the equivalent of this unit is contained within the systems software; in general, however, this unit would be required, and, as was determined later, it is needed in conjunction with the NADC CDC facility.)

3-13

At this point it was apparent that only through the introduction of additional intermediate input/output operations would the desired CDC maximum memory value be achieved, especially considering the fact that a substantial growth in central memory would occur if the technical enhancements were introduced into the program as it was then configured. Therefore, as a first step in the direction of more intermediate input/output, a change was made that permitted two fairly sizable common blocks associated with the store-search procedure to be removed from the root segment and converted to four equivalent common blocks at lower levels of the overlay structure where memory usage would be less. With this change, it appeared that the $300K_8$ goal would be achieved for the version of the code without the technical enhancements.

Concurrently with the effort to reduce central-memory requirements, the entire program was compiled on Grumman's CDC facility to identify lines of IBM code that were incompatible with CDC coding requirements. Most of the modifications that were found to be necessary involved apostrophes in format statements that had been used for expediency in the original IBM pilot code and that now were changed to the more widely accepted H format for Hollerith fields. In addition to modifications to correct compiler errors, lines of code involving double-precision operations were located, and denoted as IBM-only via the commenting procedure described previously in Subsection 3.3. For the CDC version, parallel single-precision code was introduced.

The version of the program containing both the modifications for central-memory reduction and those just discussed was the first to be transferred for debugging and checkout on Grumman's CDC facility. Included in this transfer, which was accomplished via the Grumman utility program for CDC conversion, was a substitution of the system of CDC input/output routines described in Reference 7 for the corresponding IBM routines. As discussed in that reference, both versions of these routines, which collectively have been named Disk Sequential Input Output (DSIO), were developed as a more efficient and more powerful alternative to standard Fortran input/output, and are widely used throughout both the FASTOP and FSP programs. To conserve central memory, one modification that was made to the routines of Reference 7 was to change dimension statements to correspond to a buffer size of 512 words rather than the previous value of 1024 words.

3-12

```
      SUBROUTINE AFOM (KWIT)
C
      INTEGER YES
C
      COMPLEX UMOD,VMOD
                            •
                            •
                            •

C  ******************************************************
C  *         THE FOLLOWING LINE OF FASTOP CODE HAS         *
C  *         BEEN COMMENTED OUT BECAUSE IT IS NOT          *
C  *         USED IN THE CURRENT VERSION OF ESP.           *
C  ******************************************************
C     DIMENSION ELAM(6000,3),NAMAB(2,2),NAMABT(2,2)
      DIMENSION TSH(1)               ,TSHFO(1)
      DIMENSION PHPTMP(40)
                            •
                            •
                            •

C  ******************************************************
C  *         THE FOLLOWING LINES OF FASTOP CODE HAVE       *
C  *         BEEN COMMENTED OUT BECAUSE THEY ARE NOT       *
C  *         USED IN THE CURRENT VERSION OF ESP.           *
C  ******************************************************
C
C     IF(KLUB.EQ.0) GO TO 85
C     IF(KLUQ.EQ.0) GO TO 45
C
C  (KLUQ=1) COMPUTE TRANSFORMATION MATRIX QT AND ITS TRANSPOSE Q.
C           QT=PHT*B
                            •
                            •
                            •

C     IUQT=KLUFO(1)
C      IF(IOQT.EQ.2.AND.KFREE.EQ.1) CALL PRMAT1(IUQT,IFQT,WORK,0,IUPR,7,
C     1    92,92H (TRANSPOSE OF QT TRANSFORMS DISPLACEMENTS FROM MODAL COO
C     2RDINATES TO STRUCTURAL COORDINATES))
C      IF(IOQT.EQ.2.AND.KFREE.EQ.2) CALL PRMAT1(IUQT,IFQT,WORK,0,IUPR,7,
C     1 101,101H (TRANSPOSE OF QT TRANSFORMS RELATIVE DISPLACEMENTS FROM
C     2MODAL COORDINATES TO STRUCTURAL COORDINATES))
                            •
                            •
                            •

C  85 CONTINUE
C
C  ******************************************************
C  *         END OF CODE THAT HAS BEEN COMMENTED OUT.     *
C  ******************************************************
                            •
                            •
                            •
```

Figure 3-3 - Typical Subroutine Listing Illustrating Use of Comment Cards to
Render Portions of Code Temporarily Inactive.

## 3.4 - IMPLEMENTATION

### 3.4.1 - Initial CDC Conversion

The IBM pilot code that existed at the conclusion of the effort reported
in Reference 4 was estimated to require more than $500K_8$ words of CDC central
memory, well in excess of the goal of $300K_8$ words cited above. This situa-
tion was due in part to the fact that much of the new code that was added to
the original FASTOP code to form ESP had been placed, for the sake of
expediency, in the root segment of the overlay structure. Thus, the initial
effort at reducing central-memory requirements concentrated on moving the new
routines and common blocks to lower overlay segments where possible.

Another type of change that was made to reduce central memory was to
remove from active status several regions of code, including related common
blocks, that were associated with the structural-resizing capability in FASTOP,
and therefore were not needed in ESP. The approach used here was to change the
unwanted code to comment cards, and also to introduce some additional comment
cards to mark the beginning and the end of the "commented-out" code. A typical
change of this type is partially shown in Figure 3-3. This approach was taken
to facilitate the future development of a more comprehensive program in which
the ESP and FASTOP capabilities would be combined as discussed in Section 4.

Some early ESP search code, that was logically by-passed when the search
algorithm was later refined, was also commented out. This code was retained in
the inactive form in the event that future experiences with ESP would indicate
that its reintroduction might be advantageous in some circumstances.

When it was estimated, based on IBM memory usage, that the reduction
achieved by the above changes was not sufficient to meet the $300K_8$ goal on CDC
equipment, a further refinement was introduced into the overlay structure.
Several routines and common blocks from the root segment, as well as others
from critical paths lower in the overlay structure, were moved into a second
region (equivalent to a second level in a CDC segmentation structure). Still,
the central-memory reduction was judged to be insufficient.

3-10

5.4-91 through 5.4-92a) was judged to be the best routine for the MSC interface. Next, stand-alone routines were written containing the basic NASTRAN-interface code to be used in ESP. This step was taken so that the interface procedure could be developed and checked without being encumbered by the need to update and execute the actual ESP program. Finally, the key portions of the stand-alone routines were introduced, along with appropriate logic, into the ESP routines that read the various matrices.

The development of the card-image interface capability for the dynamics-model matrices involved a modification and extension of the original FASTOP/ESP card-image input procedure that was available for the two mass matrices. Initially, an auxiliary stand-alone program was written to convert the OUTPUT2 files obtained from COSMIC NASTRAN to equivalent card-image files with the desired format. Then, code to read the flexibility and rigid-body-displacement matrices in card-image form was added to the appropriate ESP routines.

To provide the ESP user with a substantial freedom of choice in selecting combinations of input matrices, it was intended to read the four dynamics-model matrices from three different files via three input units. (The mass matrix for the free-body support point would follow the primary mass matrix in the input file for one of the units.) As will be discussed below, this approach was implemented for the MSC-NASTRAN and card-image interfaces; however, for the COSMIC-NASTRAN interface, only a single input unit was used. To further generalize the user's options in selecting ESP input matrices, logic was provided to read most possible combinations of MSC-NASTRAN, COSMIC-NASTRAN, and card-image matrices.

Since a hybrid coordinate system was used in the original ESP program (see Reference 4, Figure 4-1, page 4-2), whereas a right-hand coordinate system is used in NASTRAN, a means of resolving this difference where necessary was required. The approach chosen was to reverse the positive directions of the store lateral-translation and roll degrees of freedom in ESP. This would require relatively minor changes to the ESP code, and would preserve the traditional sign convention used in flutter analysis, i.e, nose-up rotation in pitch being considered positive when downward vertical displacement is positive.

```
          SUBROUTINE FLUTAP (KPLOTV,KPLOTF,NPLOTF)
C
          INTEGER YES
C
CIBM
C         DIMENSION BUFFER(1512)
CIBM
C
CCDC
          DIMENSION BUFFER(512)
          DIMENSION CNAME(2)
CCDC
C
          DIMENSION VBO(30), RVBO(15)
                              •
                              •
C
CCDC
          DATA CNAME/4HCALC,4HOMP /
CCDC
C
C INITIAL CONDITIONS
C
          MTAP1   =   ITAPES(37)
          CALL PROGNA (4H(FLU, 4HTAP))
          KOUNT   =   LINES
          KFIRST  =   YES
                              •
                              •
                              •
C
CIBM
C         IBUFD   =   1512
C         CALL PLOTS (BUFFER,IBUFD)
CIBM
C
CCDC
          IBUFD = 512
          ITAP60 = 60
          REWIND ITAP60
          CALL PLOTS (BUFFER,IBUFD,ITAP60)
          CALL PLOT(5.0,0.5,-3)
CCDC
C
                              •
                              •
                              •
```

Figure 3-2 – Typical Subroutine Listing Illustrating Use of Comment Cards for
                IBM-Only and CDC-Only Code – CDC Version

```
        SUBROUTINE FLUTAP (KPLOTV,KPLOTF,NPLOTF)
C
        INTEGER YES
C
CIBM
        DIMENSION BUFFER(1512)
CIBM
C
CCDC
C       DIMENSION BUFFER(512)
C       DIMENSION CNAME(2)
CCDC
C
        DIMENSION VBO(30), RVBO(15)
                        •
                        •
                        •
C
CCDC
C       DATA CNAME/4HCALC,4HOMP /
CCDC
C
C INITIAL CONDITIONS
C
        MTAP1  =  ITAPES(37)
        CALL PROGNA (4H(FLU, 4HTAP))
        KOUNT  =  LINES
        KFIRST =  YES
                        •
                        •
                        •

C
CIBM
        IBUFD  =   1512
        CALL PLOTS (BUFFER,IBUFD)
CIBM
C
CCDC
C       IBUFD = 512
C       ITAP60 = 60
C       REWIND ITAP60
C       CALL PLOTS (BUFFER,IBUFD,ITAP60)
C       CALL PLOT(5.0,0.5,-3)
CCDC
C
                        •
                        •
                        •
```

Figure 3-1 — Typical Subroutine Listing Illustrating Use of Comment Cards for
          IBM-Only and CDC-Only Code — IBM Version.

modes and the automatic-mode-elimination feature. As before, the changes were introduced with appropriate comment cards designating IBM-only and CDC-only code. This second CDC version required just under $300K_8$ words of central memory on the NADC Central Computing System, thus meeting the goal that had been set for this quantity.

### 3.4.4 - Check Cases and Execution Times

Check-out of the various modifications that were introduced was accomplished primarily by rerunning a typical abbreviated store-search case from the ESP demonstration study described in Reference 4. The data for this case was augmented/modified as appropriate, e.g., with new data for CalComp plotting, to check the various new features. In addition, data for a k-method flutter-analysis case was defined and used in test runs.

Toward the end of the check-out on Grumman's CDC facility, a case was run in which the maximum modal capacity of the program (40 modes) was tested. This run showed that computer times can be very high when a large number of modes are used: Based on approximate ratios between the Grumman Cyber 740 and the NADC Cyber 760, it is estimated that a typical search step with 40 modes might require close to one hour of Cyber 760 central processing time. The vast majority of this time is consumed by the p-k flutter-solution process, in which computer time is approximately proportional to the cube of the number of modes. Although the power of three is inherent in the solution method, the proportionality constant in this computer-time relationship is a function of the efficiency of the code. For the new version, efficiency has been reduced by both the introduction of additional intermediate input/output operations and the use of more individual read/write executions to perform some previously existing input/output operations. The use of Extended Core Storage for most new input/output has reduced the potential penalty somewhat, and the automatic-mode-elimination feature should provide some additional saving. However, the computer time required for multiple search steps with an initial number of modes close to 40 would probably still be considered excessive.

Additional information on execution times is contained in Volume I of this report, Table 4-1, page 4-5.

3-21

## 3.4.5 - New Interfaces for Dynamics-Model Data

The effort to implement a direct interface between NASTRAN and ESP began with normal-mode analyses with MSC NASTRAN using the data provided by NAVAIR. Following an initial analysis using the data received, additional runs were made with the following modifications:

(1) Inclusion of new set-definition and print case-control cards to print the modal displacements for the coordinates in the NASTRAN "analysis" set used in real eigenvalue analyses. This output was later used for comparison with the corresponding output from ESP.

(2) Introduction of case-control cards to plot the modal displacements (for easy visualization of the results).

(3) DMAP alters to compute (where necessary) and print the following matrices which are needed by ESP: the flexibility matrix; the mass matrix associated with the dynamic degrees of freedom other than those that are assumed to be fixed when computing the flexibility matrix[1]; a separate mass matrix for the degrees of freedom at the assumed free-body support point[2]; and the matrix of displacements in the dynamic degrees of freedom due to unit rigid-body displacements.

(4) Additional DMAP alters to write the above matrices to three disk files using the OUTPUT4 routine. The flexibility matrix and the rigid-body-displacement matrix were each written as separate files via different output units; the plug mass matrix followed the dynamic mass matrix in a file written via another unit. These files constituted the actual input to ESP for the MSC-NASTRAN interface.

While examining the printed output of the matrices needed for ESP, it was observed that three degrees of freedom in the NASTRAN "analysis" set had zero mass values. Since mass matrices that are used as input to ESP must be non-singular, a further modification of the NAVAIR data was made:

---

[1] For brevity, this matrix will hereinafter be called the dynamic mass matrix.

[2] Following the terminology of Reference 1, Volume I, pages 48 and 49, this matrix will be referred to as the "plug" mass matrix.

(5)  Small nonzero values that would have a negligible effect on the results
      were substituted for the original on-diagonal zero mass-matrix values.

An alternate approach, viz., using NASTRAN OMIT cards to remove the massless
degrees of freedom, could also have been used; however, the approach chosen had
the advantage of retaining all the independent degrees of freedom contained in
the original problem formulation.

After subsequent work with the COSMIC version of NASTRAN (to be described
below), two additional MSC data modifications were also found to be desirable
to facilitate comparisons of the results from the two NASTRAN versions:

(6)  Addition of a PARAM card, setting NEWSEQ equal to -1, to by-pass the
      grid-point resequencing.  (Resequencing in MSC NASTRAN was found to be
      different from that in COSMIC NASTRAN.)

(7)  Modification of the EIGR card to call out the Givens method instead of the
      modified-Givens methods for eigenvalue extraction.  (The modified-Givens
      method is not available in COSMIC NASTRAN.)

Conversion of the NAVAIR data to a form compatible with COSMIC NASTRAN
involved the following modifications to the revised MSC version of the data:

(1)  Elimination of RBAR cards by substituting CRIGD1 or CRIGD2 cards.

(2)  Substitution of BANDIT=-1 on the NASTRAN card for the PARAM NEWSEQ card to
      by-pass resequencing.

(3)  Substitution of a DIAG 21 card for the PARAM USETPRT card to print the
      grid-point and degree-of-freedom sequence numbers.

(4)  Changes to the DMAP alters to conform to COSMIC syntax and statement
      numbers.

(5)  Addition of a DMAP alter to print the eigenvectors.

(6)  The use of the COSMIC OUTPUT2 routine in place of the MSC OUTPUT4 routine
      to write the desired matrices for ESP as disk files.

For the last of these modifications, it was originally intended to make a
direct substitution of COSMIC-NASTRAN OUTPUT2 DMAP statements for the
MSC-NASTRAN OUTPUT4 statements.  This would have preserved the user option,
previously provided with MSC NASTRAN, to directly use combinations of matrices
from more than one NASTRAN run in a single ESP run.  Unfortunately, the
attempted implementation of this procedure was not successful.  Investigation

3-23

revealed that COSMIC NASTRAN, as delivered to CDC installations, provides for only one I/O unit (unit 11) in conjunction with OUTPUT2. A Fortran modification could be introduced to obtain a capability that is comparable, in terms of I/O units, to MSC NASTRAN, but it was judged that most installations would probably be using the program as received from COSMIC, and that users at these installations would prefer to restrict their OUTPUT2 usage to a single unit rather than change, or request a change to, their original NASTRAN source code. Therefore, the approach selected for using OUTPUT2 in COSMIC NASTRAN consisted of writing all four matrices desired for ESP sequentially to one disk file via unit 11.

Following the use of the modified NAVAIR data to generate the dynamics-model matrix files for ESP from both the MSC and COSMIC versions of NASTRAN, stand-alone Fortran routines were developed to read and print these files. The sample Fortran listing included with the OUTPUT4 description in Reference 6 (see pages 5.4-92c through 5.4-92f) greatly facilitated the completion of the interface code for MSC NASTRAN. In the development of the parallel code for reading COSMIC NASTRAN files, a difficulty was encountered which was subsequently traced to auxiliary information written by OUTPUT2 that is not included in the OUTPUT2 descriptions in either Reference 5, pages 5.5-24 through 5.5-27, or Reference 8, pages 4.101-1 through 4.101-3. It was determined that there is a block of header information, consisting of eight Fortran records, prior to each matrix, and, if a file rewind is performed prior to writing the first matrix, there is an additional header block of eight records at the beginning of the file. By studying octal dumps of files written by OUTPUT2, sufficient information about the header blocks was obtained to permit writing the stand-alone Fortran routine to read the OUTPUT2 files.

When the check-out of this routine was completed, a modified version of it was written to convert the OUTPUT2 files to card-image equivalents. Key portions of the stand-alone routines to read the OUTPUT2 and OUTPUT4 NASTRAN files were then introduced into the appropriate ESP routines. Also, new code to read the flexibility and rigid-body-displacement matrices in card-image form was added to the previously existing FASTOP/ESP code to read the two mass matrices in card-image form. These modifications to ESP included logic which,

3-24

except for minor restrictions, permits reading combinations of one or more matrices from each of the three types of input files (MSC-NASTRAN, COSMIC-NASTRAN, or card-image) in the same ESP run. Finally, code modifications where made to change the ESP sign convention where necessary so as to achieve consistency with the right-hand coordinate-system convention used in NASTRAN.

Although not stated thus far, the work described above relating to COSMIC NASTRAN utilized the April 1984 release of that program, since that was the version that was operational at Grumman at the time this work was done. However, when it was determined that release 17.7 was the version that was operational at NADC at that time, it was deemed prudent to run check problems using that version as well. Therefore, release 17.7 was restored to active status at Grumman, and check runs were begun.

It was determined initially that two additional changes to the COSMIC version of the NAVAIR data were required:
(1) Elimination of the BANDIT parameter on the NASTRAN card. (Resequencing for bandwidth reduction is not available as an option within COSMIC NASTRAN under release 17.7.)
(2) Conversion of all free-format data (which is not supported under release 17.7) to the standard ten fields of eight columns each.

When these changes were introduced, execution was successful until the OUTPUT2 routine was used, at which point an abnormal termination occurred. Several attempts to correct the problem met with no success, and a comparison of the OUTPUT2 source codes in the April 1984 and 17.7 releases showed that there are significant differences between the two releases in the neighborhood of the line of code at which the termination occurred. Although this tended to point toward a deficiency in the earlier version of OUTPUT2, considerable additional effort would have been required to confirm this.

Fortunately, the April 1984 release of COSMIC NASTRAN was expected to be installed at NADC only a few weeks after the release 17.7 problem was encountered. Therefore, at this point, work was shifted to introducing the new dynamics-model interface capability into the ESP version at NADC. Subsequently, when the April 1984 release of COSMIC NASTRAN became available at NADC, check

3-25

runs there of the COSMIC-NASTRAN/FSP interface were initiated and successfully completed.

## 3.4.6 - <u>User's Manual</u>

To provide information for executing FSP and interpreting the results obtained, a user's manual, Volume I of this report, was prepared. Included in this volume are instructions for preparing control-statement and input-data files, information on obtaining required dynamics-model matrices from NASTRAN, and descriptions of the ESP output. The user's manual is intended to be used in conjunction with previous FASTOP documentation, Reference 1 or 9, and to complement the theoretical description of the store-search procedure contained in Reference 4.

# 4 - CONCLUSIONS AND RECOMMENDATIONS

Although the ESP program is still a pilot code, some significant steps have been taken under this contract toward making it usable by practicing flutter analysts on modern attack-aircraft configurations. The program can now accommodate up to 40 modes including rigid-body modes, can be used either in a store-search mode or for traditional analyses, and can accept the bulk of the required dynamics-model input matrices either directly from NASTRAN or indirectly from any other upstream analysis program. Also, it is operational on the NADC Central Computing System, and a user's manual is available to permit previously uninitiated persons to prepare the required data and interpret the results.

However, as presently configured for operation on conventional CDC computing facilities, utilization of the full capabilities of the program requires very large amounts of central-processor time, most of it for the p-k flutter-solution procedure. Since considerable computing inefficiency exists in this area, due to the large number of intermediate input/output operations that are being employed to reduce central-memory requirements, there is a potential for significantly reducing the time usage from the present level. One minimal step that might be taken is to expand the utilization of Extended Core Storage to include areas beyond the new intermediate input/output that was added during this contract. Additionally, further program restructuring, to provide more central memory for the p-k solution procedure in exchange for new intermediate input/output in other less-critical portions of the code, might be advantageous. Also, despite the added constraints on the computing-system job mix that can be run at the same time that ESP is executing, a net gain in throughput might be achievable by allowing central-memory usage to increase to a greater percentage of the physical maximum than is presently used.

Since the modifications just cited are less-than-ideal approaches to reducing the currently large ESP execution times, a preferred approach might be to look beyond the memory restrictions of most current CDC installations, and to assume instead that future ESP usage will be primarily either on new larger CDC scalar machines, e.g., the Cyber 845, or on large vector processors,

e.g., a Cray or the Cyber 205. The greatly increased memory available in these machines not only will assure that the full performance potential of ESP can be achieved, but it also will minimize the effort needed to achieve improved performance. Of course, the use of a vector processor would have the advantage of substantially reducing central-processor times as well as permitting the desired elimination of most intermediate input/output operations.

Beyond the subject of computing-efficiency improvements for the p-k flutter-solution procedure, additional tasks are recommended to convert ESP from its present pilot-code status to a more user-oriented production code. One such task is the introduction of better output titles and annotation, such as providing units for all quantities, to improve the readability of listings. Also, to verify that all applicable and desirable FASTOP options are also operational in ESP, an extensive set of check problems should be formulated and run, and, if necessary, corrective coding changes should be introduced.

To realize the full potential of ESP for expediting the attack-aircraft development process, a long-term objective should be to fully integrate ESP with the most recent generally available version of FASTOP (Reference 9). This would permit automated determination of flutter-critical configurations via ESP, followed by automated composite- or metallic-structure minimum-weight resizing in FASTOP to achieve required flutter speeds. Repetition of this process as necessary, until the last configurations for which resizing is performed are also the critical configurations from the next ESP pass, would provide the desired near-optimum distribution of structural material for the required external-store combinations.

# 5 - REFERENCES

1.  Wilkinson, K. et al., "An Automated Procedure for Flutter and Strength Analysis and Optimization of Aerospace Vehicles", AFFDL-TR-75-137, Vols. I and II, December 1975. (Final report for Air Force Flight Dynamics Laboratory contract F33615-72-C-1101)

2.  Chipman, R. R., and Malone, J. B., "Study of a Method for Determining Critical Store Configurations for Wing-Store Flutter", Grumman Aerospace Corporation Report No. ADCR-76-1, September 1976. (Final report for Naval Air Systems Command contract N00019-76-C-0160).

3.  Chipman, R. R., and Malone, J. B., "Application of Optimization Technology to Wing/Store Flutter Prediction", Journal of Aircraft, Vol. 15, No. 11, November 1978, pp. 786-793.

4.  Chipman, R. R. and Laurie, E. J., "Demonstration of a Method for Determining Critical Store Configurations for Wing-Store Flutter", Grumman Aerospace Corporation Report No. ADCR-80-1, May 1980. (Final report for Naval Air Systems Command contract N00019-79-C-0062)

5.  Anon., "The NASTRAN User's Manual (Level 17.5)", NASA SP-222(05), December 1978.

6.  McCormick, C. W., ed., "MSC/NASTRAN User's Manual", MacNeal-Schwendler Corporation Report No. MSR-39, February 1981.

7.  Markowitz, J. and Stylianos, P., "Documentation of and Improvements to the CDC DSIO System in FASTOP-3", Grumman Aerospace Corporation Report No. ADCR-79-1, October 1979. (Final Report for Air Force Flight Dynamics Laboratory contract F33615-79-M-3226)

8.  Anon., "The NASTRAN Programmer's Manual (Level 17.0)", NASA SP-223(04), December 1977.

9.  Markowitz, J., and Isakson, G., "FASTOP-3: A Strength, Deflection and
    Flutter Optimization Program for Metallic and Composite Structures",
    AFFDL-TR-78-50, Vols. I and II, May 1978.   (Final report for Air
    Force Flight Dynamics Laboratory contract F33615-77-C-3027).

# END

# FILMED

5-85

# DTIC